**TT HOLDING DATABASE DESIGN REPORT**

**Table of Contents**

(make it)

**List of Figures**

* *Figure 1.0 Crow’s foot ERD.*
* *Figure 1.1 Creation of database and first 4 tables.*
* *Figure 1.2 Creation of next 3 tables continued.*
* *Figure 1.3 insertion of 10 rows in first 3 tables.*
* *Figure 1.4 insertion of next 2 tables.*
* *Figure 1.5 insertion on last 2 tables.*
* *Figure 2.0 creation of 2 views.*
* *Figure 3.0 first 2 insert triggers.*
* *Figure 3.1 4 triggers (2 for update and 2 for delete).*
* *Figure 3.2 trying to insert a duplicate phone number and Preventing Salary Updates Below a Minimum Value.*
* *Figure 4.0 calling the function and get the total number of employees in the person table.*
* *Figure 4.1 function getting the total number of organizations in the organization table.*
* *Figure 5.0 The add\_person procedure and add\_organization procedure.*
* *Figure 5.1 creation of 2 users and granting them privileges.*

**List of Tables**

* + Person
  + Organization
  + Position
  + Position\_Held
  + Employment
  + Permanent\_Employee
  + Part\_Time\_Employee
  + organization\_view
  + position\_held\_view

**List of Abbreviations**

* SQL – Structured Query Language
* ERD – Entity Relationship Diagram
* RDBMS – Relational Database Management System
* FK – Foreign Key
* PK – Primary Key
* DDL – Data Definition Language
* DML – Data Manipulation Language
* DCL – Data Control Language
* UML – Unified Modeling Language
* DBMS – Database Management System

**Abstract**

A database is information that's set up for easy access, management and updating. Computer databases typically store aggregations of data records or files that contain information such as sales transactions, customer data, financials and product information.

Databases are used for storing, maintaining and accessing any sort of data. They collect information on people, places or things. This information is gathered in one place so it can be observed and analyzed. Databases can be thought of as an organized collection of information.

Databases are essential for storing large amounts of data in one place. With databases, organizations can quickly access, manage, modify, update, organize and retrieve their data (Alexander 2024).

The TT Holding Database Design project’s aims to develop a database system that effectively manages employment relationships, positions, and organizations. The system ensures efficient tracking or retrieving of employees working across multiple organizations, whether as permanent or part-time employees. The report details the design, implementation, and testing of the system using MySQL, incorporating ER diagrams, and advanced SQL features such as triggers, stored procedures, and views. Additionally, it covers data insertion, validation, and testing to ensure integrity and performance.

**Chapter 1: Introduction**

**1.1 Problem Statement**

Organizations need a structured database to manage employees who may work in multiple organizations and hold multiple positions over time. The lack of an efficient system leads to difficulty in tracking employment history, salaries, and job roles.

**1.2 Problem Solving**

This project introduces a relational database system that efficiently models employment, organizations, and job positions while ensuring data integrity through foreign key constraints.

it’s important to understand that employee database management provides your business with a central location from which to consolidate your employee data. Rather keeping this data siloed across numerous discrete platforms, an employee database management system allows you to generate reports on your workforce in just a few clicks, while you’ll also be able to access the data when you’re out of the office or outside normal working hours. (GoCardless,2021)

**1.3 Objective**

* To design an ER/EER diagram mapping employment relationships.
* To create and implement MySQL database tables with relationships.
* Data insertion between 10 and 20 rows per table.
* To apply advanced SQL techniques such as views, triggers, functions, and procedures.
* To ensure user privileges and data security.

**1.4 Scope & Constraints**

* **Scope**: Covers employment tracking, organization relationships, and job positions.
* **Constraints**: Limited to MySQL implementation without external database integrations.

**Chapter 2: Literature Review**

**2.1 Introduction**

This chapter looks at previous studies on database design, relational database management systems (RDBMS), and employment tracking databases. It focuses on the best ways to organize data about employees and organizations.

**2.2 Reviewing the Literature**

**Review of relational database models for employment systems:**

Fundamentals of Database Systems (Elmasri & Navathe, 2020)

* This book explains ER modeling and database relational schema design, which were applied in designing the TT Holdings Database to avoid redundancy and maintain data integrity.

SQL Triggers and Stored Procedures in Database Security (Lee, 2022)

* The research highlights the use of SQL triggers and stored procedures for data security and automation. This approach was applied by implementing triggers for insert, update, and delete operations, ensuring data consistency.

Foreign key Constraint

* Foreign key constraints are used to link the columns of two tables together, ensuring the referential integrity of the data. In essence, a foreign key in one table points to a primary key in another table, indicating that the rows in these two tables are related (2024 FK).
* Advanced SQL techniques in database optimization.

**2.3 Findings & Discussion**

* The literature suggests that relational databases effectively manage employment relationships.
* EER diagrams improve data representation for complex relationships.
* Advanced SQL features (triggers, procedures) enhance data integrity and automation.

**Chapter 3: Methodology**

All SQL commands used for creating tables, triggers, functions, and procedures are documented in a separate notepad file, which is included in the appendices (Appendix 0). This file serves as a reference for the full list of SQL statements executed during the development of the TT Holding Database System.

**3.1 Requirement Analysis**

* Identification of system requirements.
* Understanding user needs for employment and organization tracking.
* Defining database schema and relationships.

**3.2 System Design**

**3.2.1 Architectural Design**

**Example of Tables and Relationships together with their cardinality:**

|  |  |  |
| --- | --- | --- |
| **ENTITIES** | **RELATIONSHIP** | **CARDINALITY** |
| Person – Employment | Employment | Many-to-Many |
| Employment –Organization | Have employees | Many-to-Many |
| Person – Position\_Held | Hold position | Many-to-Many |
| Organization – Position | Have positions | One-to-Many |
| Person – Permanent\_Employee | Be Permanent | One-to-One |
| Person – Part\_Time\_Employee | Be Part timed | One-to-One |

**3.2.2 UML Diagrams**

**Entity relationship diagram (ERD)**

An Entity Relationship Diagram (ER Diagram) pictorially explains the relationship between entities to be stored in a database. Fundamentally, the ER Diagram is a structural design of the database. It acts as a framework created with specialized symbols for the purpose of defining the relationship between the database entities. ER diagram is created based on three principal components: entities, attributes, and relationships (Haroon 2025).

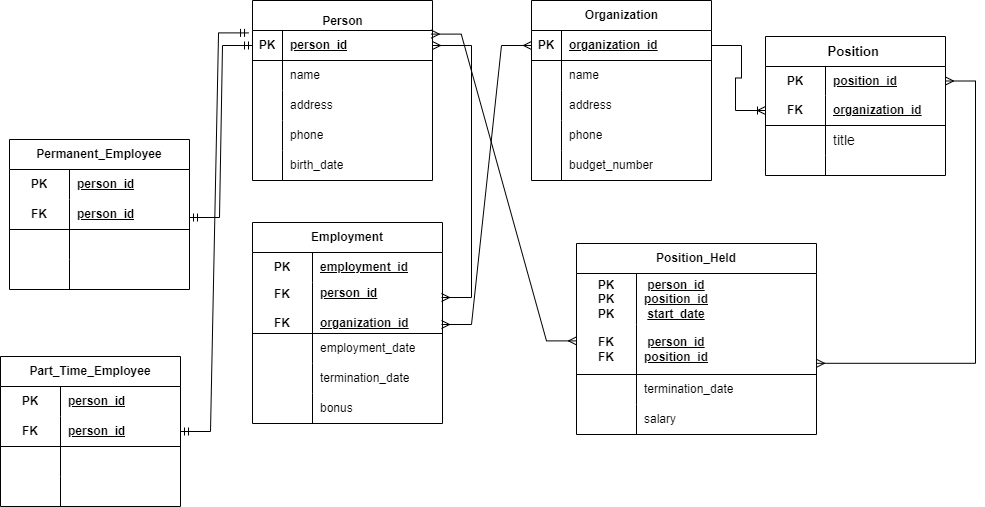
**Use of ER Diagrams in DBMS:**

* ER Diagram helps you conceptualize the database and lets you know which fields need to be embedded for a particular entity
* ER Diagram gives a better understanding of the information to be stored in a database
* It reduces complexity and allows database designers to build databases quickly
* It helps to describe elements using Entity-Relationship models
* It allows users to get a preview of the logical structure of the database.

**The following diagram showcases multiple entities related to employment tracking:**

1. The "Person" entity represents individuals with attributes such as person\_id, name, address, phone, and birth\_date.
2. The "Organization" entity contains details of various organizations, including organization\_id, name, address, phone, and budget\_number.
3. The "Position" entity stores job positions with attributes position\_id, title, and a foreign key linking it to an organization.
4. The "Employment" entity captures relationships between a person and an organization, including employment\_id, employment\_date, termination\_date, and bonus.
5. The "Position\_Held" entity establishes a many-to-many relationship between Person and Position, tracking historical employment records with attributes position\_id, person\_id, start\_date, termination\_date, and salary.
6. The "Permanent\_Employee" and "Part\_Time\_Employee" entities are specializations of "Person."

**Crow’s Foot ER Diagram showing relationships:**



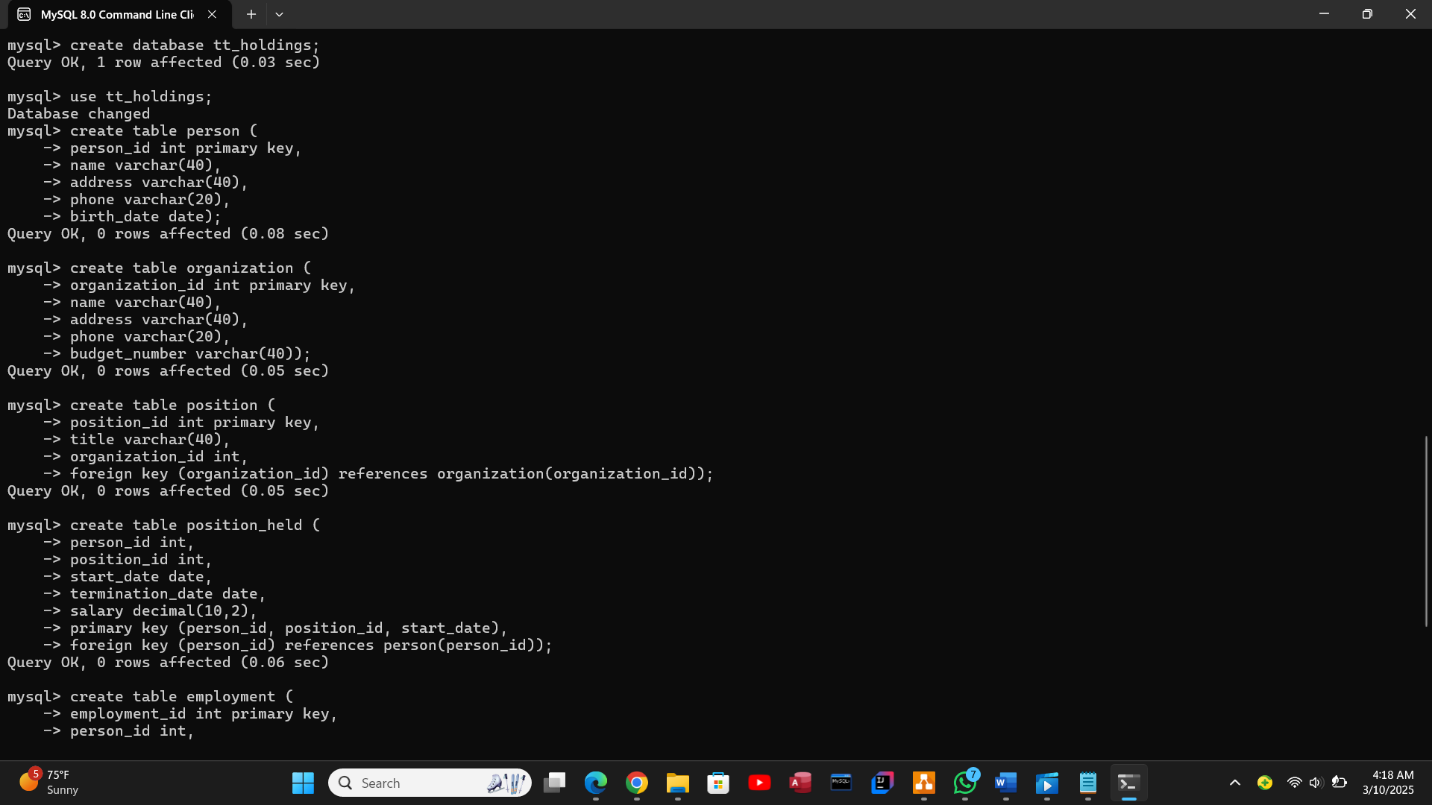
*Figure 1.0 Crow’s foot ERD.*

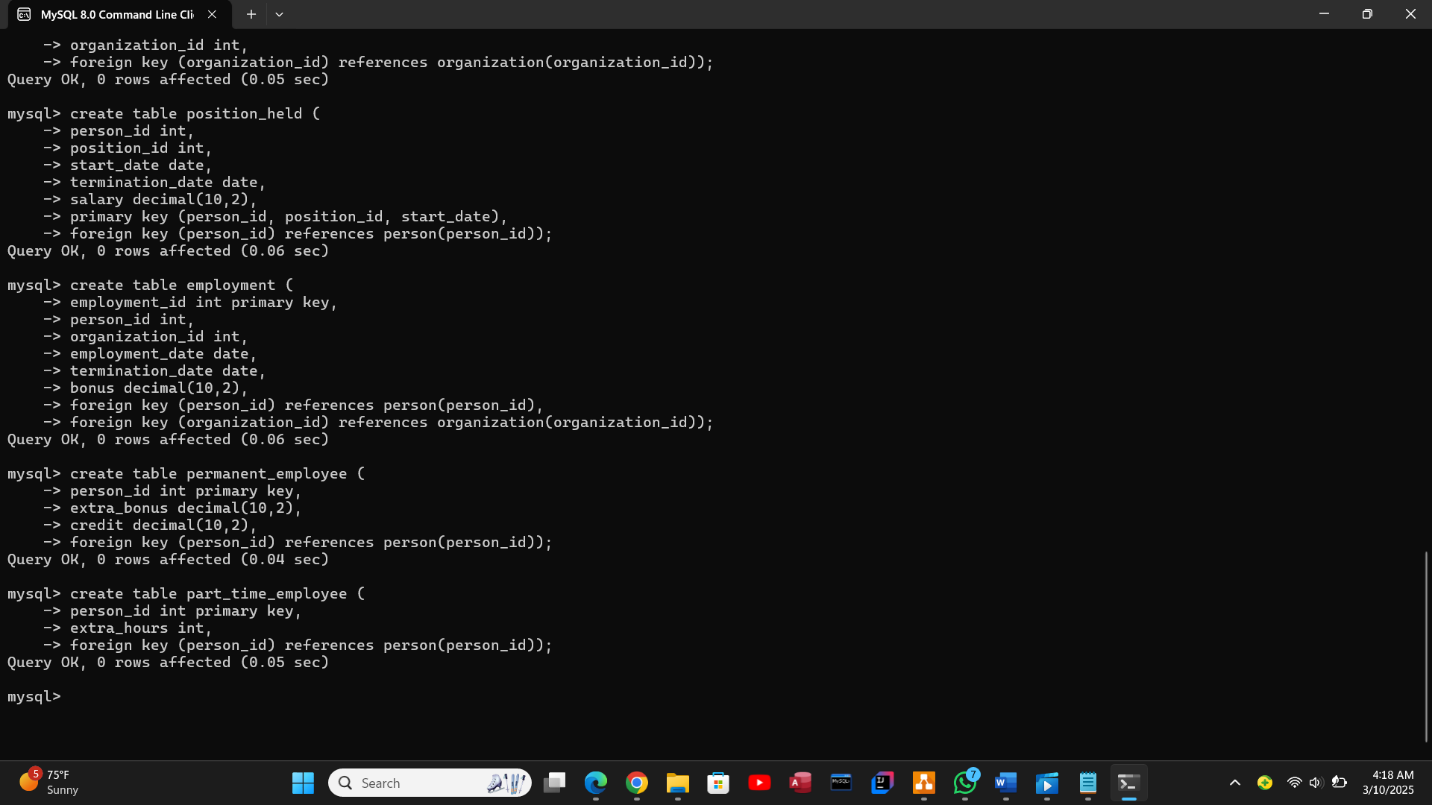
**3.3 System Implementation / Prototyping**

MySQL database implementation

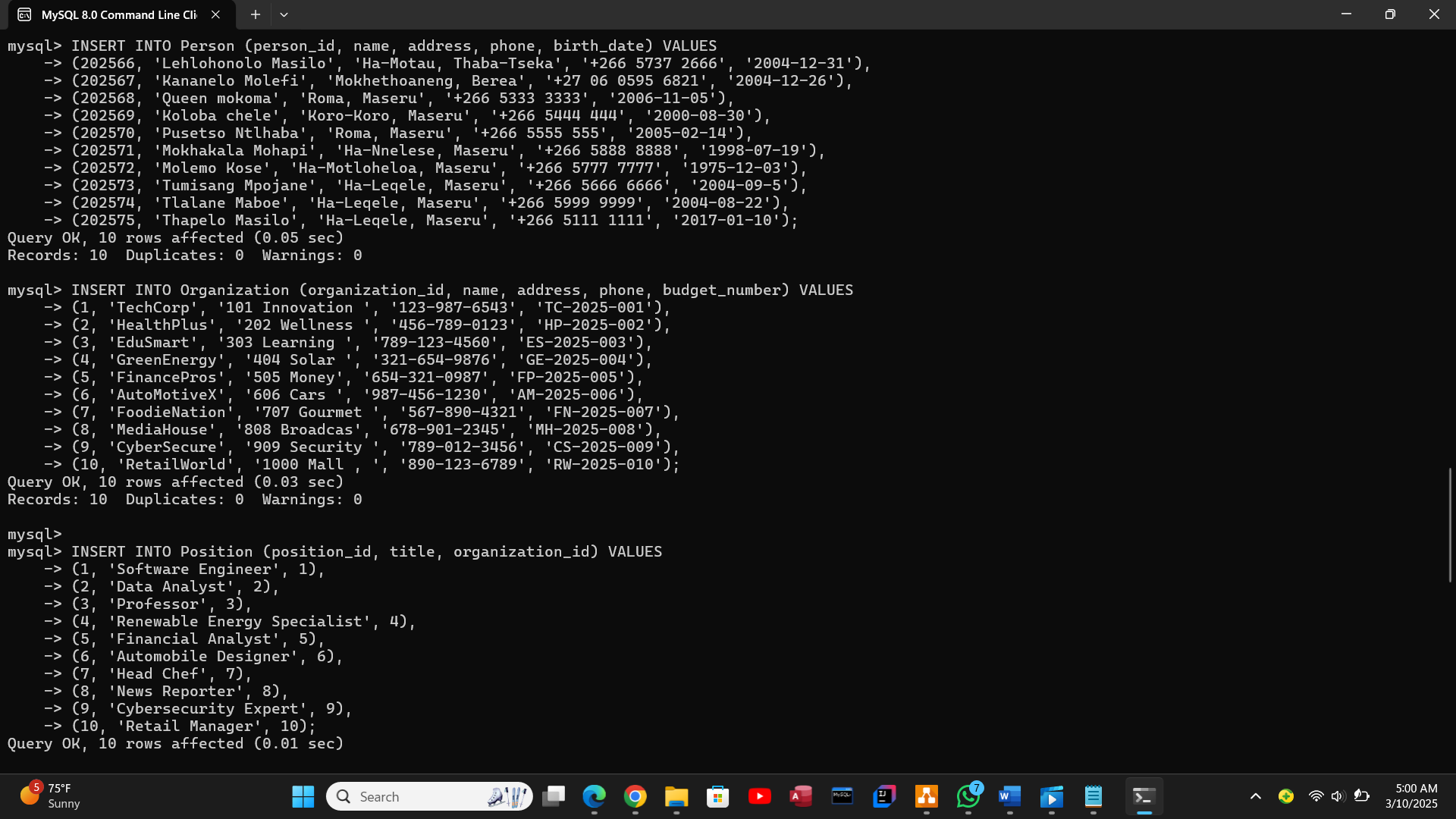
The TT\_HOLDINGS database was created using MySQL, following a relational database model. Incorporating various tables, views, triggers, functions, and stored procedures to ensure data integrity and efficient data retrieval. Below are the steps taken in implementing advanced SQL features.

**Creation of Database and tables**

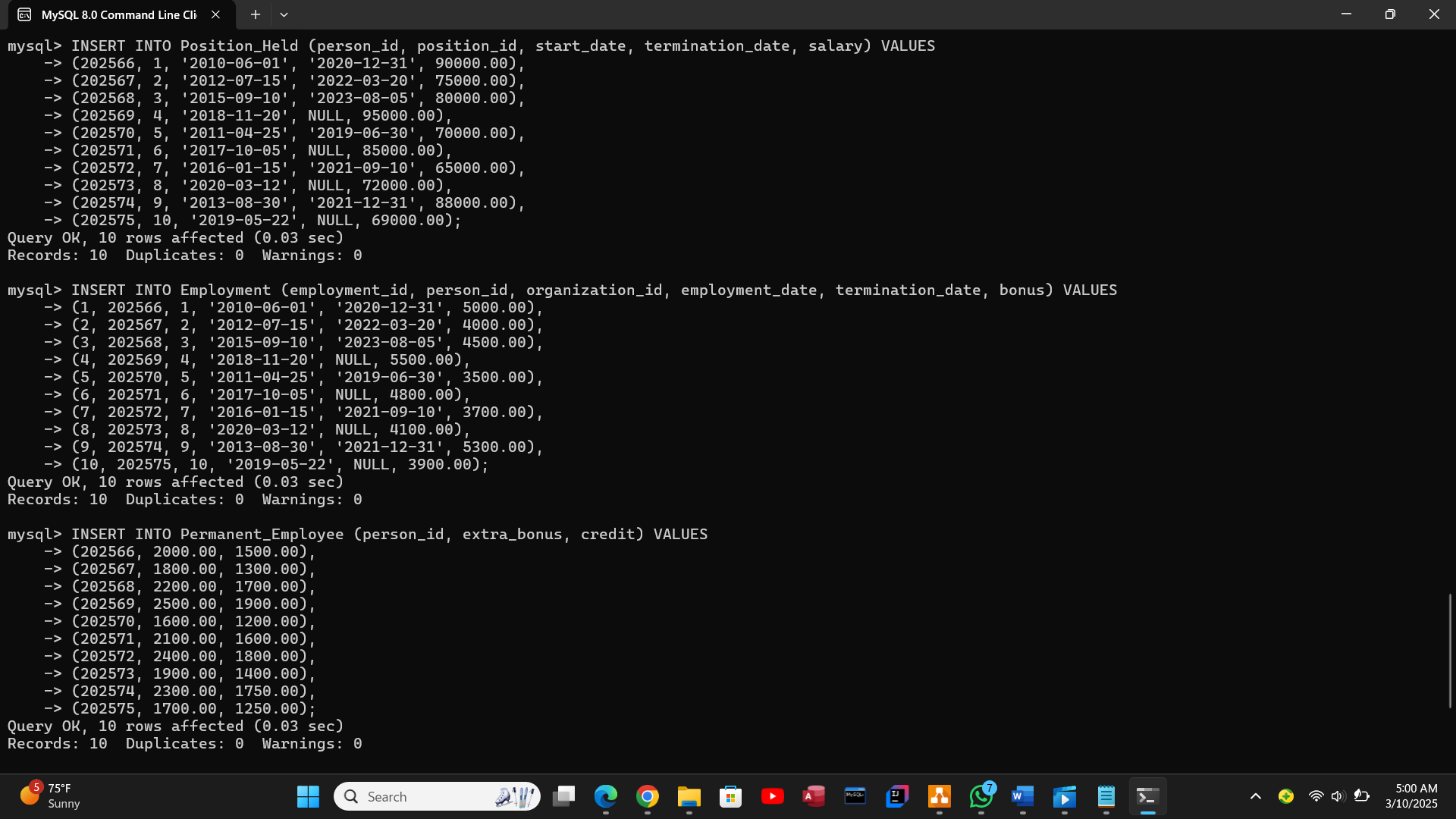


*Figure1.1 Creation of database and first 4 tables.* 

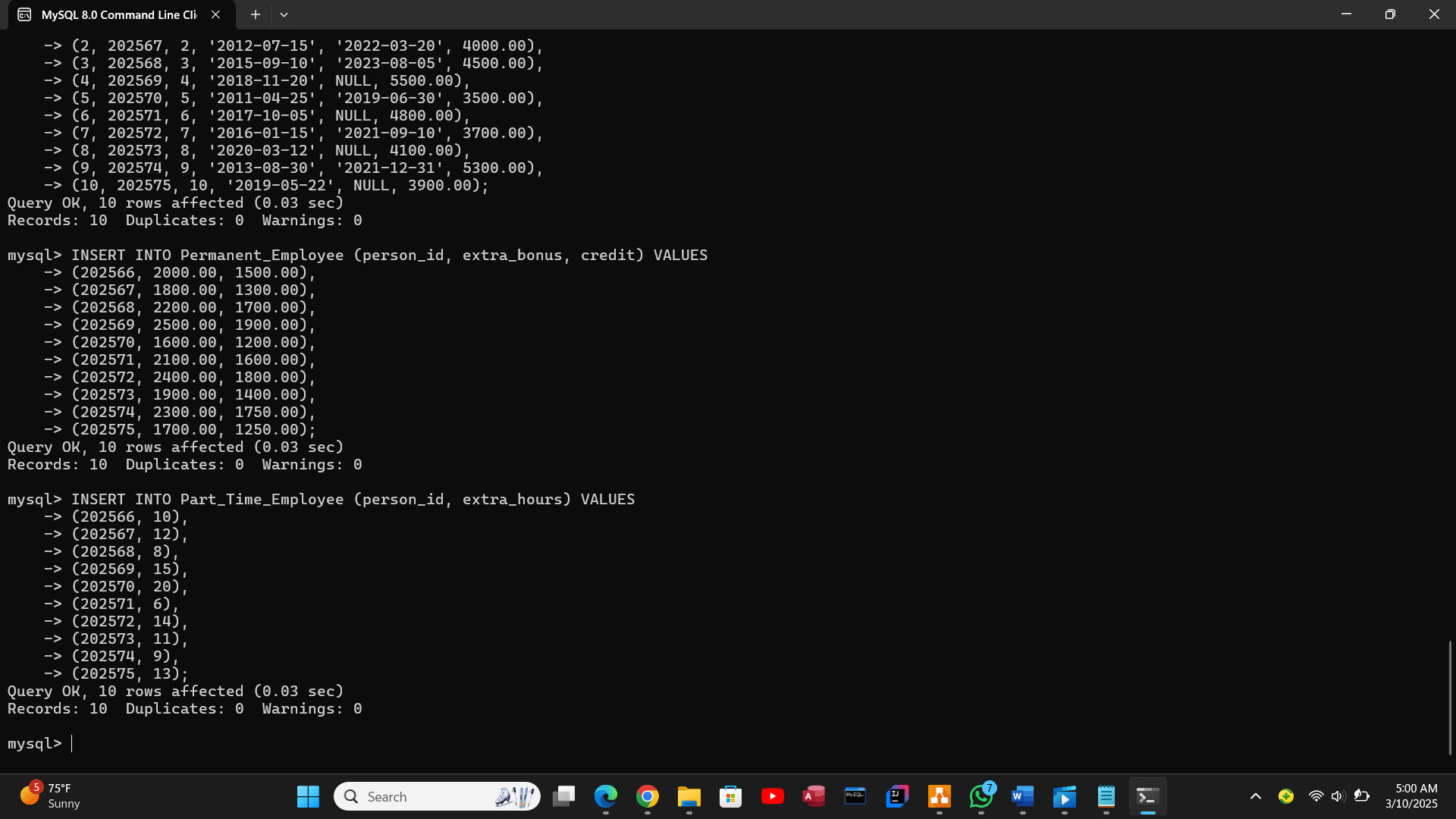
*Figure 1.2 Creation of next 3 tables continued.*

**Data insertion**

*Figure 1.3 insertion of 10 rows in first 3 tables.*



*Figure 1.4 insertion of next 2 tables.*

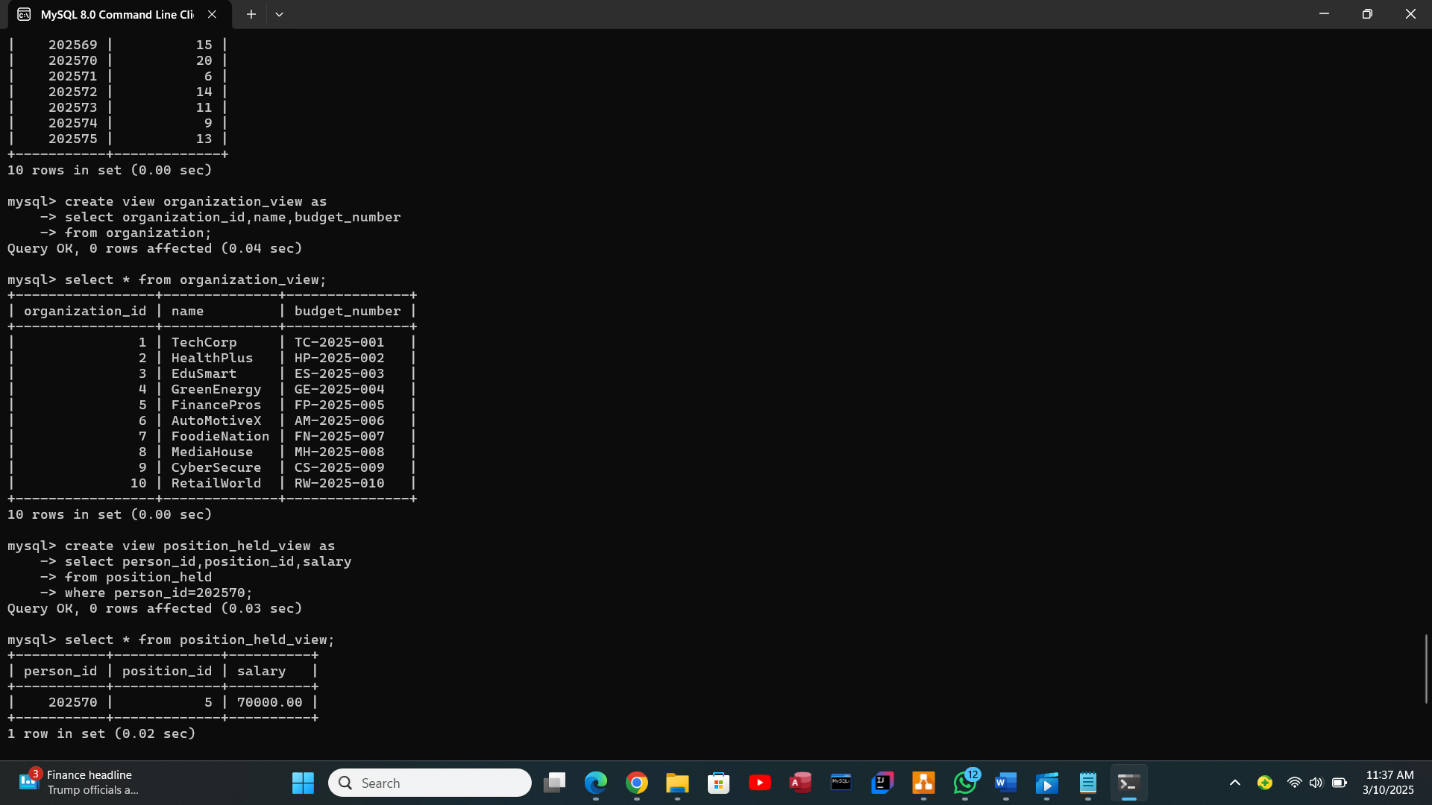


*Figure 1.5 insertion on last 2 tables.*

**Advanced SQL (Views, Triggers, Functions and 2 Procedures).**

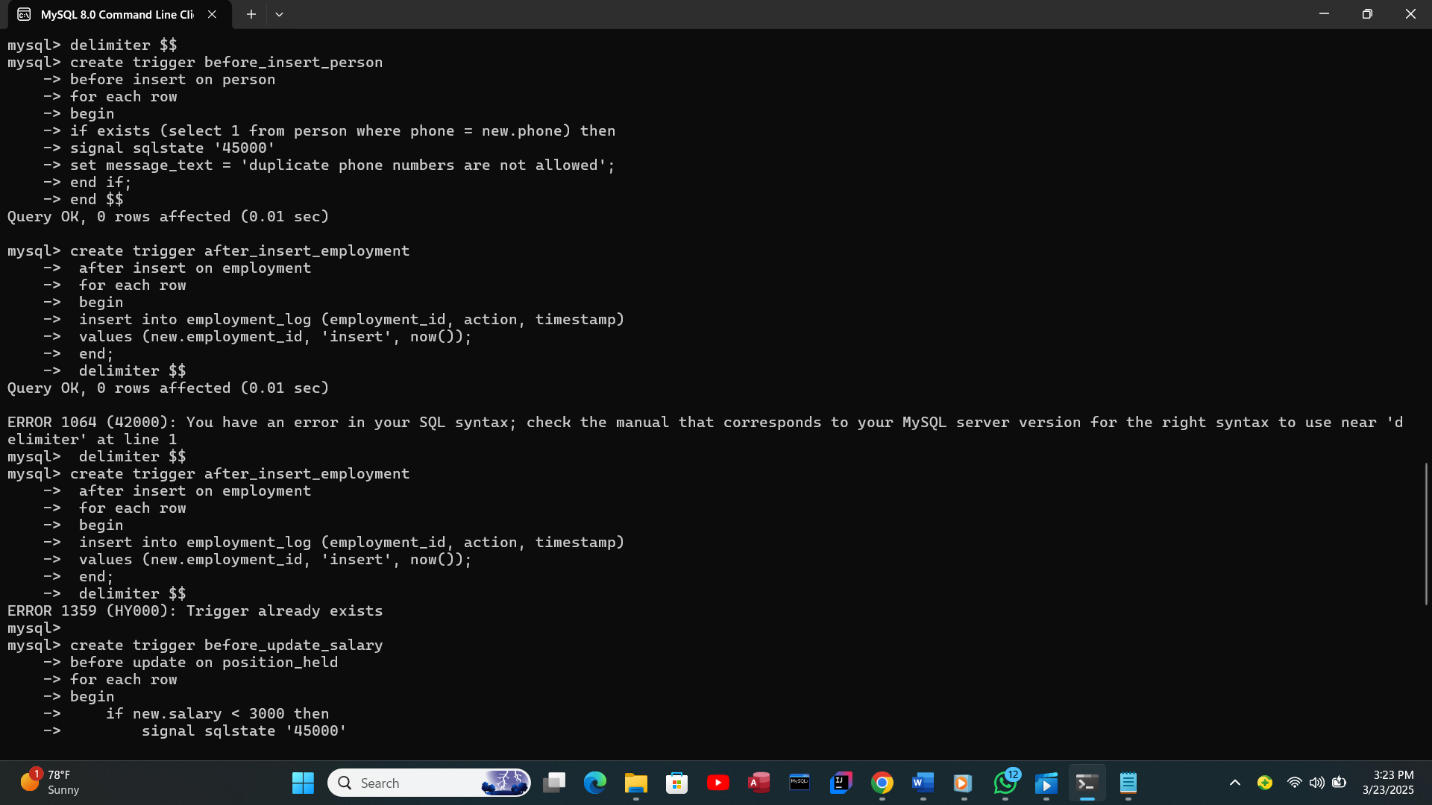
**Views creation**

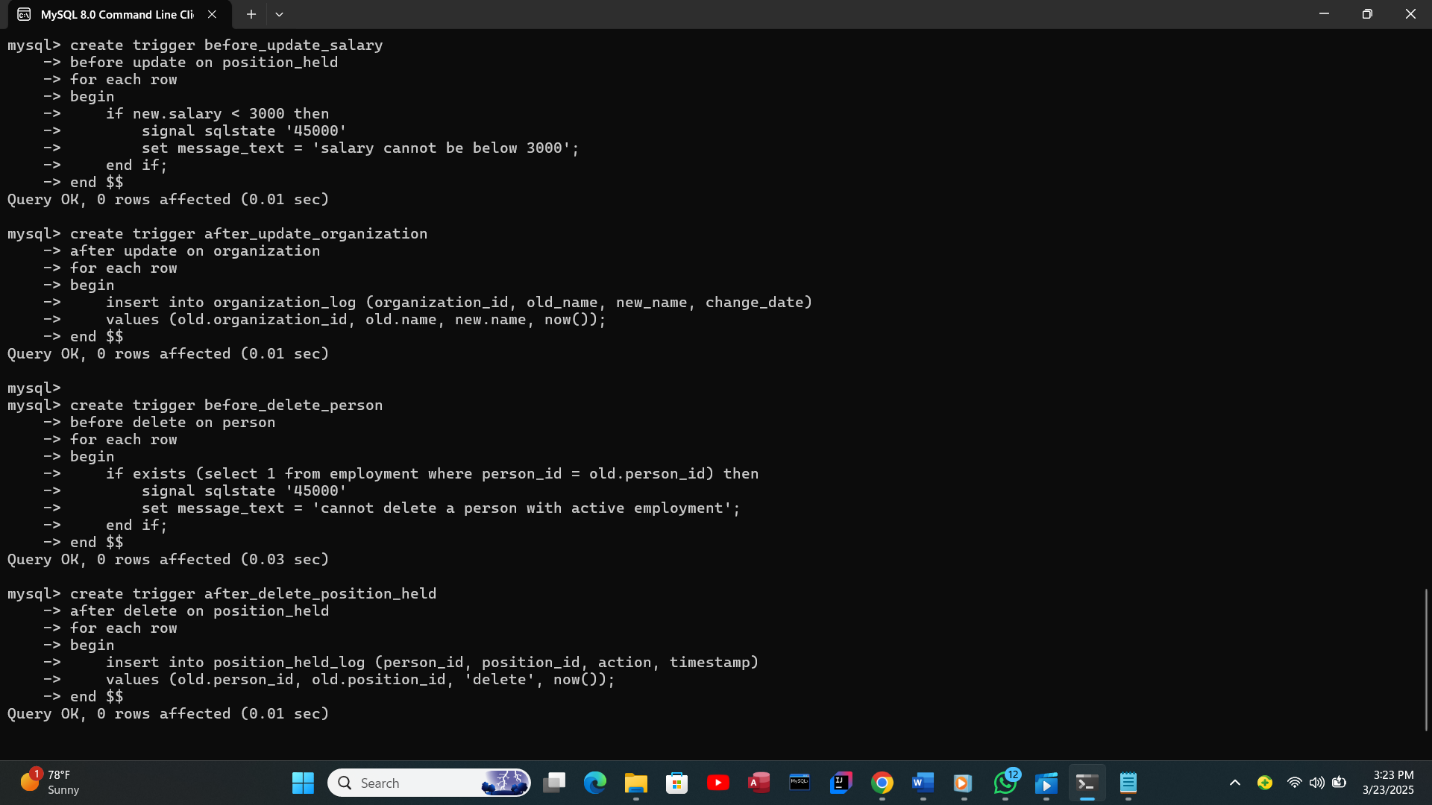
Views are a special version of tables in SQL. They provide a virtual table environment for various complex operations. You can select data from multiple tables, or you can select specific data based on certain criteria in views. It does not hold the actual data; it holds only the definition of the view in the data dictionary (Avinash 2025).



*Figure 2.0 creation of 2 views.*

**Triggers Implementation**

A trigger in SQL is a set of instructions that are executed (or “fired”) in response to a specific event on a particular table or view in a database. This event can be an action like an insert, update, or delete operation. The primary purpose of triggers is to maintain the integrity of the data in the database and impose certain business logic or rules (Pranav 2023).*Figure 3.0 first 2 insert triggers.*

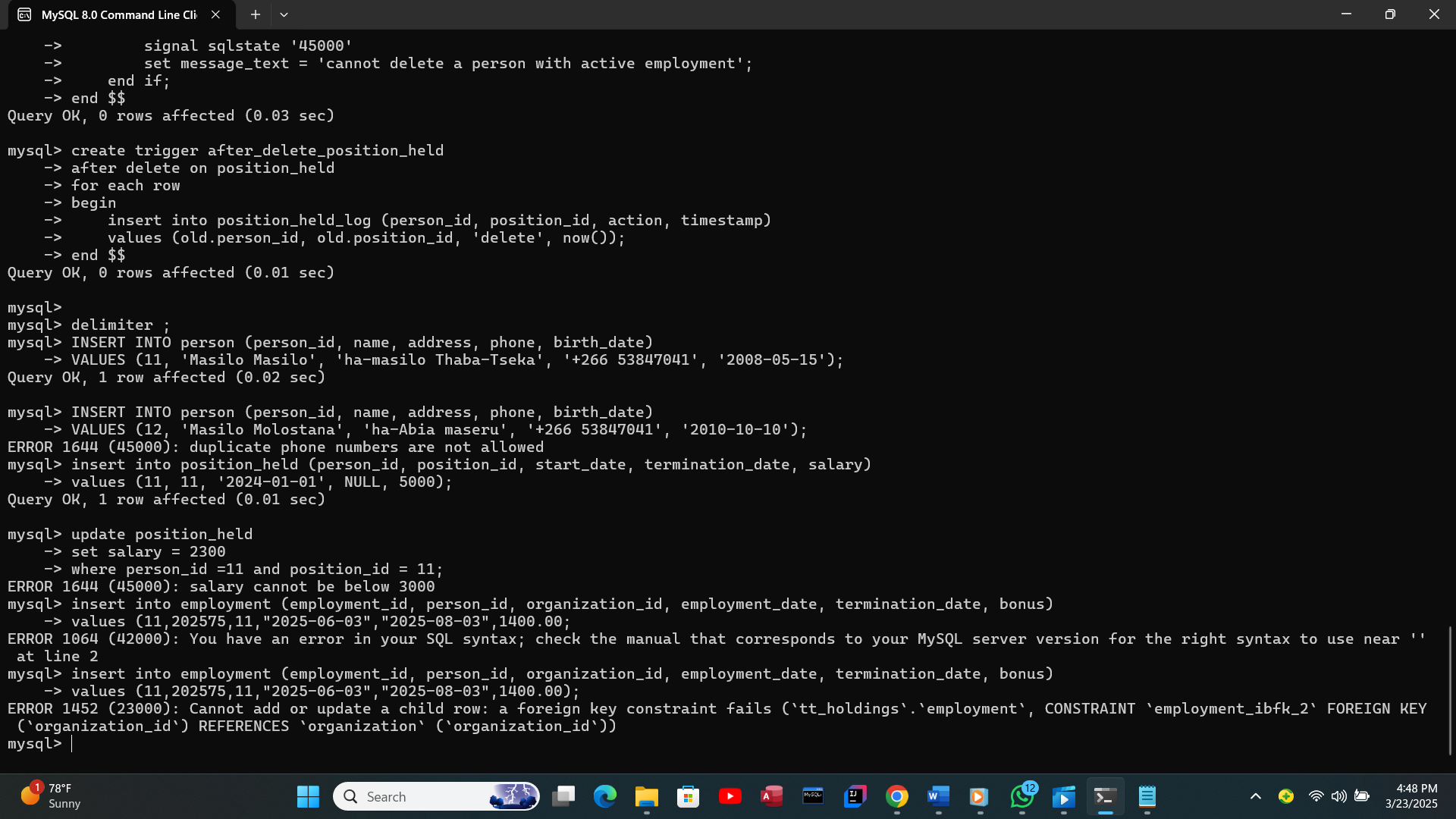
**

*Figure 3.1 4 triggers (2 for update and 2 for delete).*

**3.4 Testing**

Unit testing for individual tables by inserting values into them. Sample data was inserted into each table and data insertion is 10 rows per table.

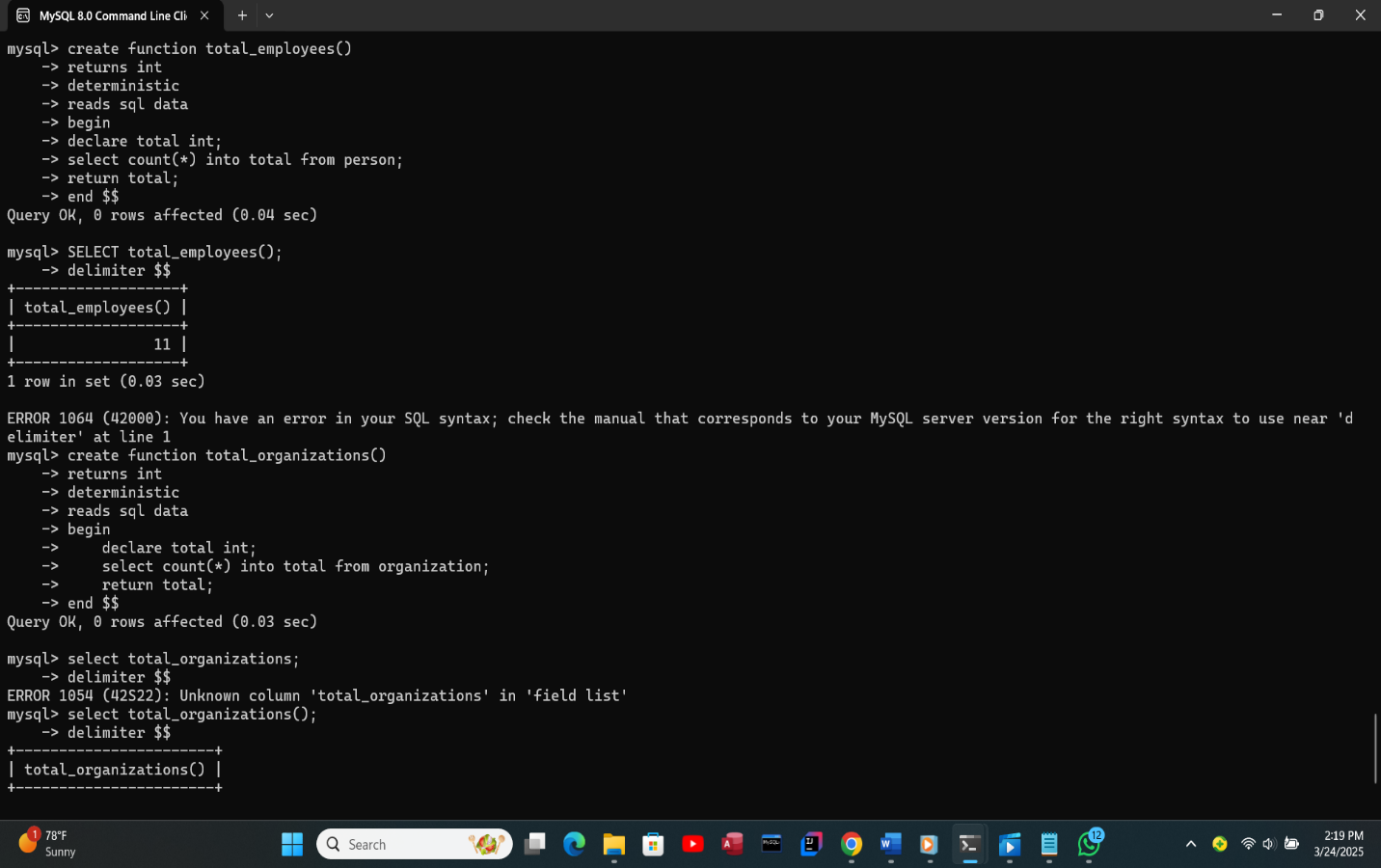
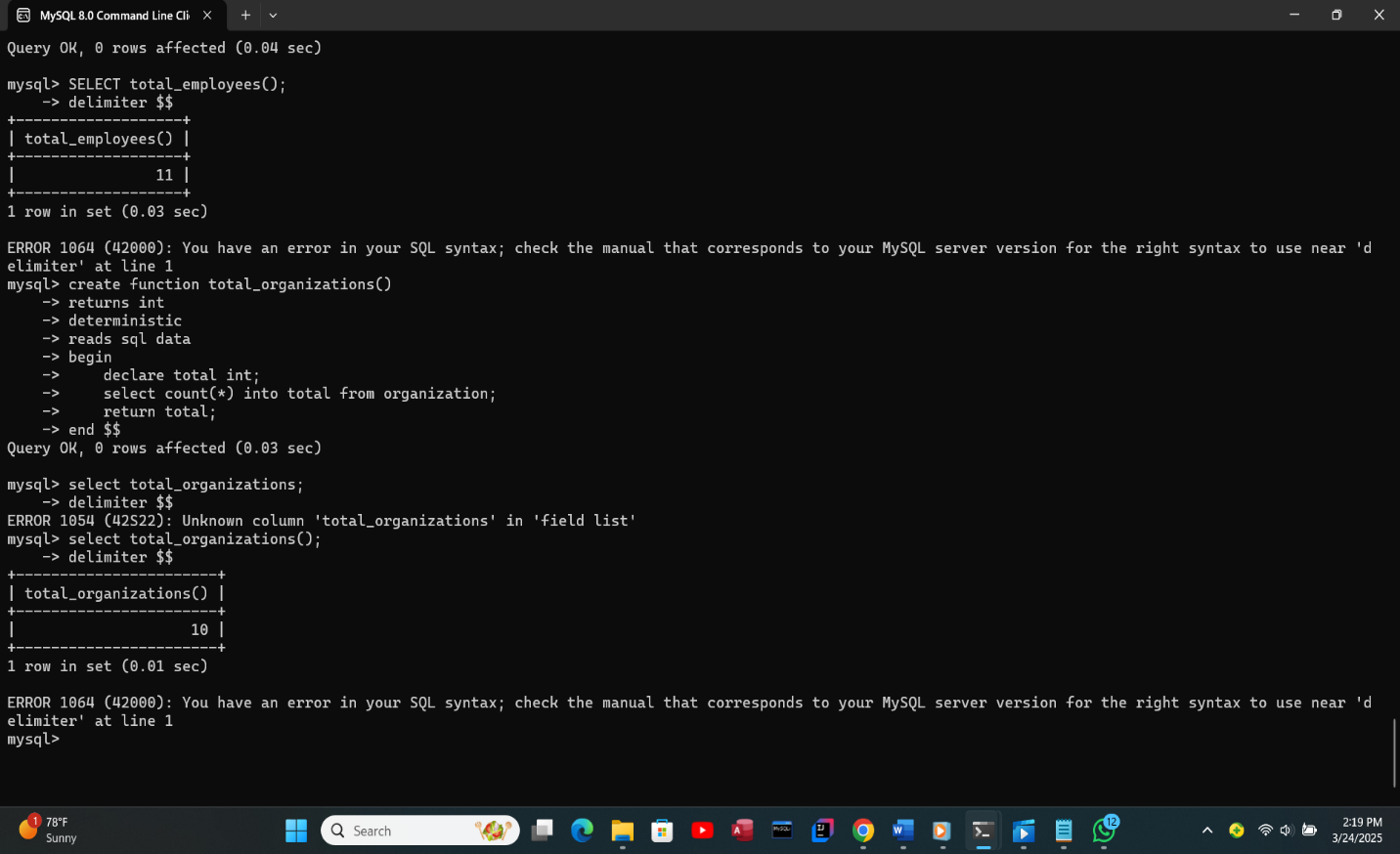
**Test Triggers by Performing Relevant Actions**

****

*Figure 3.2 trying to insert a duplicate phone number and Preventing Salary Updates Below a Minimum Value.*

**Functions**

Function in SQL servers are database objects that include a group of SQL statements to perform a specified activity. A function takes parameters, performs actions, and returns the outcome. It should be noted that functions always return either a single value or a table (Shailendra Chauhan 2025).

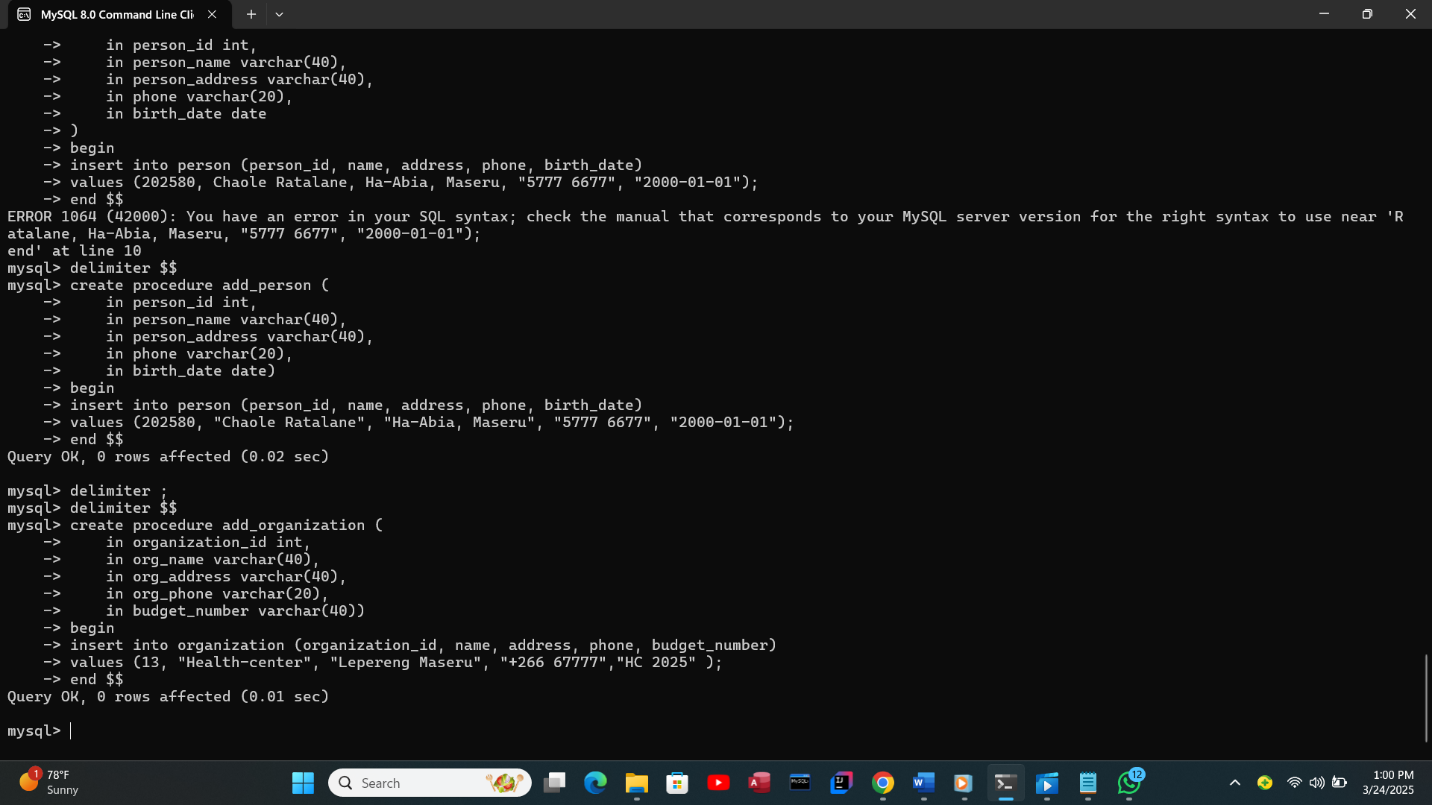
*Figure 4.0 calling the function and get the total number of employees in the person table.*

*Figure 4.1 function getting the total number of organizations in the organization table.*

**Procedures**

A stored procedure in SQL is a group of SQL statements that are stored together in a database Based on the statements in the procedure and the parameters you pass, it can perform one or multiple DML operations on the database, and return value, if any. Thus, it allows you to pass the same statements multiple times, thereby, enabling reusability (Rakivirun 2024).

The add\_personprocedure allows the insertion of a new person into the database while ensuring proper data handling and add\_organization procedure simplifies adding organizations by using structured SQL queries.

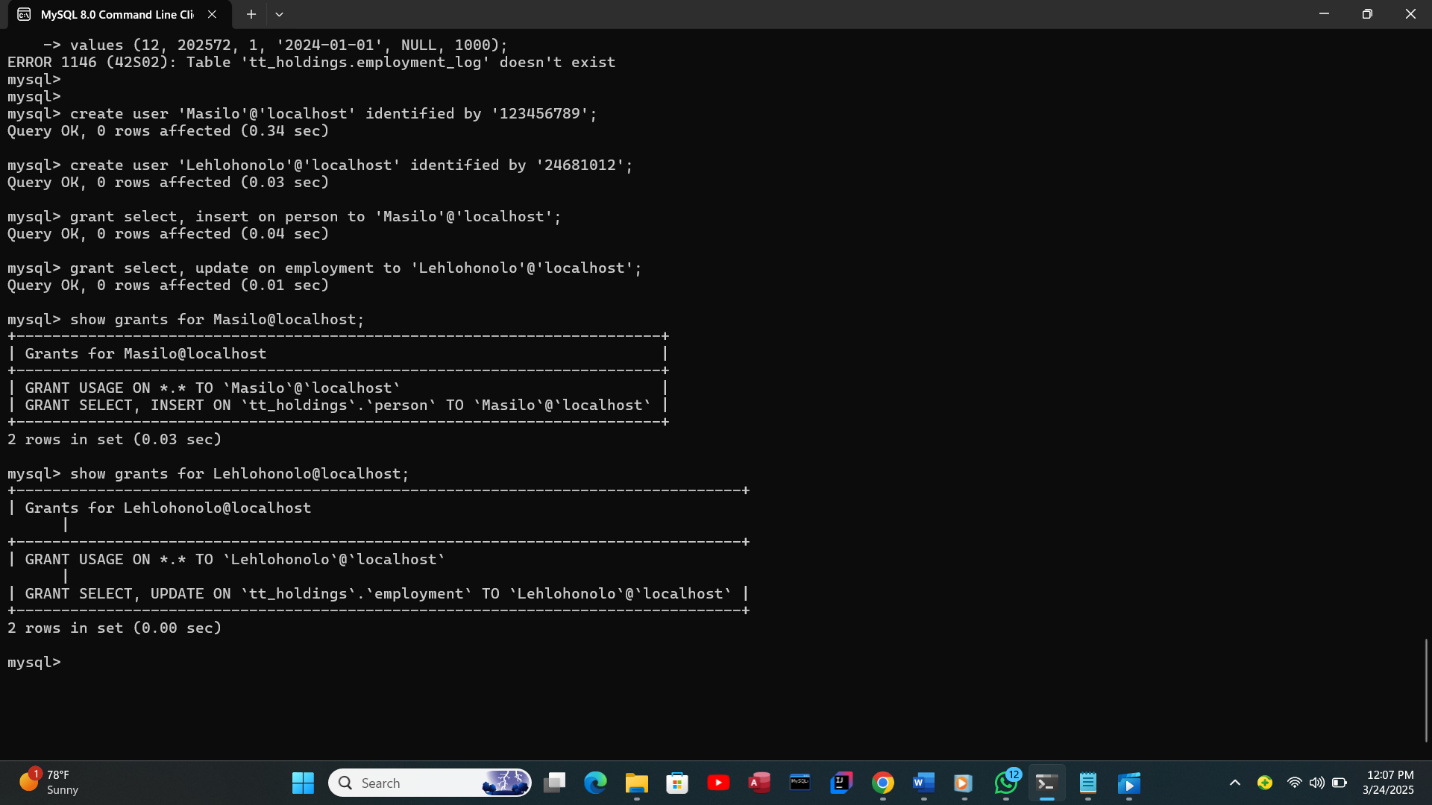


*Figure 5.0 The add\_person procedure and add\_organization procedure.*

**User Creation and Privilege Assignment**

SQL users are accounts in a database system that allow authenticated access to resources. SQL create user command is used to create new users in a database system. Users are identified by unique usernames and passwords. The command allows specifying the username and password for the new user. Once created, users can be assigned specific privileges to control their access and actions within the database. This command is essential for managing user accounts and ensuring data security (geekforgekks 2024).

**The SQL commands used for this process are shown below:**



*Figure 5.1 creation of 2 users and granting them privileges.*

**Chapter 4: System Initiation and Planning**

**4.1 Assessing Project Feasibility**

Technical Feasibility:

* The database is designed using MySQL, a widely supported relational database management system (RDBMS).
* The system follows a structured schema with relationships between Person, Organization, Employment, and Position tables.

Economic Feasibility:

* MySQL is open-source, eliminating licensing costs.
* Hardware and storage requirements are minimal for small to medium-sized organizations.

Operational Feasibility:

* The database structure supports efficient employee and organization tracking.
* The system is user-friendly with simple SQL queries for data management.

1. **Resource used**
   1. Software Tools: MySQL, ER Diagram tool (Draw.io).

**4.2 Project Plan**

**Development Timeline and Milestones**

|  |  |  |  |
| --- | --- | --- | --- |
| **Phase** | **Tasks** | **Duration** | **Milestone** |
| **Planning & Analysis** | Define requirements, feasibility study | 1 day | Requirements finalized |
| **Design** | Create ERD, relational schema, and mappings | 2 hours | ERD and schema completed |
| **Implementation** | Create tables, relationships, and constraints | 2 days | Database structure completed |
| **Data Insertion** | Insert test data (10-20 rows per table) | 1,5 days | Data successfully inserted |
| **Advanced SQL Development** | Create views, triggers, functions, and procedures | 1 day | SQL components implemented |

**Chapter 5: System Analysis**

**5.1 Determining System Requirements**

**Identifying Key Data Entities**

* **Person** – Stores employee details.
* **Organization** – Represents companies or institutions.
* **Position** – Defines job roles within an organization.
* **Position\_Held** – Records employment history for each person.
* **Employment** – Tracks employment details between a person and an organization.
* **Permanent\_Employee** – Specialized table for permanent employees.
* **Part\_Time\_Employee** – Specialized table for part-time employees.

**Functional Requirements**

* The system should allow adding, updating, and deleting employee records.
* It should maintain a history of employees’ positions and salaries.
* It must support different employment types (permanent and part-time).

**5.2 Structuring System Requirements**

**Database Schema and Relationships**

* Each table has a primary key for unique identification.
* Foreign keys establish relationships between tables, enforcing referential integrity.

**Chapter 6: Conclusion**

**6.1 Advantages of the System**

* Efficient tracking of employment relationships.
* Data consistency and integrity.

**6.2 Future Enhancement of the System**

* Integration with external HR systems.
* Development of a user-friendly interface.

**6.3 Potential Benefits**

* Improved workforce management.
* Streamlined database operations.

**6.4 Conclusion**

This report presented the complete design and implementation of the TT Holding Database System, ensuring efficient data management for employment tracking. Future enhancements will focus on automation and user accessibility.

**References**

(To be added)

**Appendices**

Appendix 0: SQL Commands

* A notepad file is included, containing all the SQL commands used to create the database, tables, functions, triggers, procedures, and any other commands executed during the project. This file serves as a comprehensive record of all the steps taken to implement the TT Holding Database System.

Appendix 1: SQL Scripts

* Database Creation Script
* Table Creation Scripts
* Data Insertion Queries
* Views Creation Queries
* Trigger Implementation Scripts
* Stored Procedures and Functions

Appendix 2: ER Diagrams and UML Diagrams

* Crow’s Foot ERD
* Employment Relationship UML Diagram

Appendix 3: Test Cases and Validation

* Test cases for data insertion and retrieval
* Testing triggers

Appendix 4: User Creation and Privilege Assignment

* SQL Queries for User Creation
* Privilege Assignment Queries